INTERNATIONAL JOURNAL OF ADVANCE RESEARCH AND DEVELOPMENT

IJARnD

(Volume3, Issue4)

Available online at: www.ijarnd.com

Technology challenge white paper: Coping with spool space challenge in Teradata

Rajneesh Shukla

Solution Architect, IT Professional

ABSTRACT

Teradata is a high performance database system. It uses spool space to perform queries on the database. It writes the result from each step in the query execution plan to disk, to a spool file and subsequently reading these results. Hence it is important to plan spool space utilization in optimized way, else we will end up with spool space error. Spool space and capacity planning are mutually dependent concepts. It is important to increase the system's performance without making any hardware changes. It is challenge as data is continuously increasing and most systems will respond to increased load with some degree of decreasing performance. Considering the importance a lot of work has been done all over the world on DWH performance tuning, it is critical to understand multiple facts to be considered to ensure optimal utilization of Teradata Spool Space. If we are using the Spool Space in an optimal way then we are increasing the system performance without causing any additional cost to organization. Poor planning will result out of spool space memory and hence can lead to failure of the Teradata based application. Understanding of spool space, its allocation, causes of spool space error, mitigation plan are detailed in this white paper.

Keywords: *Teradata, Spool Space, Dataware house, Technology Challenge.*

1. INTRODUCTION

Spool is temporary disk space used to hold intermediate rows during query processing, and to hold the rows in the answer set of a transaction.

Spool space is allocated to a user as a total amount available for that user, however this total amount is spread across all AMPS. For example, 100AMP system, a user with 10G spool limit means: 100M spool/AMP Spool space is a shared resource with limits placed upon users at the Profile or User level. Furthermore, spool space for a given user is shared across ALL active sessions. Spool is an attribute of the user account you are using to connect to the Teradata environment; it is not really an attribute of the database itself. Each database users may have a “spool limit” that restricts the user to allocate more spool area at a time, than its limit. Keep in mind that all active sessions of a username must share the spool limit together. Once the query is complete, the space is released. It only holds data. It is active up to the current sessions only. If there is no limit defined for a particular database or user, limits are inherited from parents. The best way to estimate spool usage is to run explains on all of your queries. Just remember that this is an estimate of spool usage not the actual spool usage. Spool space is divided by the number of AMPs. Whenever per AMP limit exceeds, the user will get a spool space error. Teradata recommends 20% of the available perm space is allocated for Spool space varies across applications. It is quite challenging to ensure query execution is following optimal path with respect to spool space because Spool Space utilization is impacted by many factors. The objective of this report is to detail out various ways to utilize Spool Space in optimal way to improve the overall system performance, get rid of spool space errors without causing any additional cost to organization.

2. TERADATA SPACE CONCEPTS

There are three types of spaces available in Teradata:

Permanent Space	Spool Space	Temp Space
Permanent space is the maximum amount of space available for the user/database to hold data rows. Permanent tables, journals, fallback tables and secondary index sub-tables use permanent space. Permanent space is not pre-allocated for the database/user. They are just defined as the maximum amount of space the database/user can use. The amount of permanent space is divided by the number of AMPs. Whenever per AMP limit exceeds, an error message is generated.	Spool space is the unused permanent space which is used by the system to keep the intermediate results of the SQL query. Users without spool space cannot execute any query. Similar to Permanent space, spool space defines the maximum amount of space the user can use. Spool space is divided by the number of AMPs. Whenever per AMP limit exceeds, the user will get a spool space error.	Temp space is the unused permanent space which is used by Global Temporary tables. Temp space is also divided by the number of AMPs.

There are 4 types of spool spaces available in Teradata:

Intermediate	Output	Persistent	Volatile
Intermediate spool results are retained until no longer needed. You can determine when intermediate spool is flushed by examining the output of an EXPLAIN.	Output spool results are the final information returned for a query or the rows updated within, inserted into, or deleted from a base table.	When Redrive protection is enabled, Teradata Database stores responses for sessions that participate in Redrive in non-fallback persistent spool tables. Persistent spools are not deleted following a Teradata restart or node failure. Persistent spools are retained until the SQL request completes and the application has fully received the response.	The system uses volatile spool space for volatile tables. This is necessary because volatile tables do not have a persistent stored definition.

Sources of Spool Space:

Spool space is taken only from disk cylinders that are not being used for data. Data blocks and spool blocks cannot coexist on the same cylinder.

When spool is released, the file system returns the cylinders it was using to the free cylinder list

3. SPOOL SPACE ERROR- CAUSES and SOLUTION

Spool space is allocated to a user as a total amount available for that user, however this total amount is spread across all AMPS. This means if you're allocated 200G of spool and you're on a system with 24 AMPS, you're allocated ~8.3G of spool on each AMP. If any AMP exceeds its allocated amount of spool during a transaction you submit, your query is cancelled with error: NO MORE SPOOL SPACE.

For example, all the rows in the query are being processed on a few amps creating a "hot amp" situation. In this case, just a few amps are racking up spool while the others sit comparatively idle. This is caused when the tables in the query are missing stats, have been improperly Primary Indexed, or are otherwise "untuned".

Cause	Solution
COLLECT STATISTICS	**Missing, poor, or stale/aged statistics may lead to below issues:** • Duplicates a large spool/table to all AMPS Thinks it is small. • Redistributes spool/table by a skewed value Thinks it is non-skewed. **Facts to take care:-** • We should be careful to not over collect on statistics. • If a table is updated by several inserts multiple times a day, the statistics do not need to be refreshed after each insert. • One collection is Significant after the last insert. • Statistics are not needed for temp tables that are not joined to other tables and only used for staging.

	• We should not collect statistics on empty tables. • This will cause the optimizer to choose an inefficient path based on the information available to the parser. • For tables being completely refreshed, the statistics are needed after the refresh. • Statistics should be collected when a table is initially loaded and anytime the table's demographics change by more than 10%. **COLLECT STATS should include:-** • Individual columns in an index. • All columns in an index, multi-column where size is less than 16 bytes. • Join columns. • Filter or qualifying columns. • Secondary Index.
PRIMARY INDEX	Primary index is used to specify where the data resides in Teradata. It is used to specify which AMP gets the data row. Each table in Teradata is required to have a primary index defined. If the primary index is not defined, Teradata automatically assigns the primary index. Primary index provides the fastest way to access the data. A primary may have a maximum of 64 columns. A poor primary index having lumpy distribution data which can cause a query to run several hours when it should execute in seconds/minutes. Hence, we should choose a single column or multiple columns that distribute the data evenly across all AMPS. Eliminate columns from the primary index that have a lot of null values. Value change rate should be low or never. Column(s) should be frequently used in join constraint. If the table joins to a similar table having the same columns, the primary index on both tables should be the same.
SKEWING	Teradata skewing can be considered one of the worst problems on any Teradata system. A high skew factor means in effect, the parallelism of the system is degraded leading to: • Poor CPU parallel efficiency on full table scans and bulk inserts. The AMP holding the most records of the many values will be the bottleneck, forcing all other AMPs to wait. • Increased IO for updates and inserts of biased values, considering the extra workload for the AMP with a high number of multiple rows for the same NUPI value. Facts to consider: • Proper primary index specification should evenly distribute the rows of a table across the AMPs. This prevents skewing. • The Query Log Information in SQL Assistant and other Editors tells us about the degree of Skewing in a query. CPU Skew > 50 reflects worse case scenarios and generally any query having a CPU Skew > 4 is considered poor performing. Hence, by seeing the CPU Skew from the Query Log Information a programmer can easily make out which query needs to be fine-tuned to avoid high Skew.
SQL JOIN	**PRODUCT JOIN:** This is a cross join every row from one table is joined to every row on the second table. Spool file is as large as (No. Of rows table one * No. Of rows table two), large product joins (billions of rows) should be avoided. Product joins are most efficient when a SMALL lookup table is duplicated. Product joins are

	inefficient when large fact tables are duplicated (this can indicate aged or missing stats). **MERGE JOIN:** Requires sort of spool files. Merge join are efficient when there is not a many to many relationship on columns involved in the join. If there is a many to many relationship, try to aggregate the columns on one table to reduce the volume by creating a volatile table, derived table or work table. **HASH JOIN:** The tables do not have to be sorted and the smaller table can be much larger than for a product join. The smaller table/spool is "hashed" into memory. Then, the larger table is scanned and for each row, it looks up the row from the smaller table in the hashed table that was created in memory. If the smaller table is broken into partitions to fit into memory, the larger table must also be broken into the same partitions prior to the join.
MULTISET OR SET TABLE	A set table performs a duplicate row check. If there are a lot of non-unique values for a primary index, this can be very CPU intensive. For example, for a primary index having 2000 values a duplicate row check will be performed 4,000,000 times. This is referred to as chaining. The first record is loaded. The next record having the same PI value to load, checks all the columns of the first one to determine if it is a duplicate. Once the third record is loaded, it checks both the first and the second records and so on. A Multiset table allows duplicate rows so the duplicate row check is omitted. If duplicates can be omitted using a group by or filtered programmatically, a load to a multiset table performs better. A Multiset table having a NUPI, non-unique primary index, with occurrences between 500 – 2000 is not bad. For tables having non unique primary index where there are several hundred or a couple of thousands values for a given primary index ‘use a multiset table’ For tables having a more unique index like 1 to 10 values for a give primary index ‘use a set table’
SQL CODE	Check if we can add filters or additional joins to reduce volume. The casting – both implicit and explicit – are consuming CPU that is not necessary. When a typecast is applied to a column the optimizer cannot use the existing statistics anymore, because it is different data types. Additionally it also cannot use an existing index on that column. A execution plan should be analyzed properly and thoroughly and reason for no confidence should be identified. Queries having derived tables will often show no confidence because the optimizer does not know how many rows are in a derived table. Use pre-computed aggregate values at data modification time, rather than at select time. Minimizing number of Index, which help in saving space and reduces DML queries execution time. Minimizing need of joins. Minimize foreign keys on tables. Know the resource usage by getting results like below: 1) Total CPU Usage 2) Spool Space needed 3) The LHR (Ratio between CPU and I/O usage) 4) CPU Skew 5) Skew impact on CPU

4. CONCLUSION

There are multiple causes which may lead to spool space error like all the rows in the query are being processed on a few amps creating a “hot amp” situation (just a few amps are racking up spool while the others sit comparatively idle), missing stats, have been improperly Primary Indexed’d, the query is excluding critical join criteria resulting in cartesian products, untuned sqls, over statistics etc. It may be also a case that you just don’t have enough space to perform the query.

The only way to know for certain is to look at the EXPLAIN plan for your query.

However knowing the genuine causes and applying proper solutions to eliminate those issues will help in achieving better performance, smooth functioning of Teradata based Dataware House solution without adding additional hardware cost to Organization.

5. REFERENCES

[1] https://www.tutorialspoint.com/teradata/teradata_space_concepts.htm
[2]https://info.teradata.com/HTMLPubs/DB_TTU_16_00/index.html#page/Database_Management/B035-1094-160K/jjn1472240614273.html
[3] https://www.google.com/